

前端理论及术语

Doctype作用是什么？

声明叫做文件类型定义（DTD），声明的作用为了告诉浏览器该文件的类型。让浏览器解析器知道应该用哪个规范来解析文档。声明必须在 HTML 文档的第一行，这并不是一个 HTML 标签。

二、严格模式与混杂模式如何区分？它们有何意义？

严格模式：又称标准模式，是指浏览器按照 W3C 标准解析代码。

混杂模式：又称怪异模式或兼容模式，是指浏览器用自己的方式解析代码。

如何区分：浏览器解析时到底使用严格模式还是混杂模式，与网页中的 DTD 直接相关。

- 1、如果文档包含严格的 DOCTYPE，那么它一般以严格模式呈现。（严格 DTD ——严格模式）
- 2、包含过渡 DTD 和 URI 的 DOCTYPE，也以严格模式呈现，但有过渡 DTD 而没有 URI（统一资源标识符，就是声明最后的地址）会导致页面以混杂模式呈现。（有 URI 的过渡 DTD ——严格模式；没有 URI 的过渡 DTD ——混杂模式）
- 3、DOCTYPE 不存在或形式不正确会导致文档以混杂模式呈现。（DTD不存在或者格式不正确 ——混杂模式）
- 4、HTML5 没有 DTD，因此也就没有严格模式与混杂模式的区别，HTML5 有相对宽松的语法，实现时，已经尽可能大的实现了向后兼容。（HTML5 没有严格和混杂之分）

意义：严格模式与混杂模式存在的意义与其来源密切相关，如果说只存在严格模式，那么许多旧网站必然受到影响，如果只存在混杂模式，那么会回到当时浏览器大战时的混乱，每个浏览器都有自己的解析模式。

三、严格模式与混杂模式的语句解析不同点有哪些？

1) 盒模型的高宽包含内边距padding和边框border

在W3C标准中，如果设置一个元素的宽度和高度，指的是元素内容的宽度和高度，而在IE5.5及以下的浏览器及其他版本的Quirks模式下，IE的宽度和高度还包含了padding和border。

2) 可以设置行内元素的高宽

在Standards模式下，给span等行内元素设置width和height都不会生效，而在quirks模式下，则会生效。

3) 可设置百分比的高度

在standards模式下，一个元素的高度是由其包含的内容来决定的，如果父元素没有设置高度，子元素设置一个百分比的高度是无效的。

4) 用margin:0 auto设置水平居中在IE下会失效

使用margin:0 auto在standards模式下可以使元素水平居中，但在quirks模式下却会失效,quirk模式下的解决办法，用text-align属性:

```
body{text-align:center};#content{text-align:left}
```

5) quirk模式下设置图片的padding会失效

6) quirk模式下Table中的字体属性不能继承上层的设置

7) quirk模式下white-space:pre会失效

补充内容:

一、常用的具体声明:

1、HTML5 (一种) :

2、HTML 4.01 (三种) : 严格模式包含所有 HTML 元素和属性，但不包括展示性的和弃用的元素 (比如 font)，不允许框架集 (Framesets) ; 过渡模式包含所有 HTML 元素和属性，包括展示性的和弃用的元素 (比如 font)，不允许框架集 (Framesets) ; 框架模式等同于过渡模式，但允许框架集内容。

```
HTML 4.01 Strict : <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

```
HTML 4.01 Transitional : <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

```
HTML 4.01 Frameset : <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

3、XHTML 1.0 (四种) : 前三种模式同上，XHTML 必须以格式正确的 XML 来编写标记。

```
XHTML 1.0 Strict : <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
XHTML 1.0 Transitional : <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
XHTML 1.0 Frameset: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

```
XHTML 1.1 该 DTD 等同于 XHTML 1.0 Strict, 但允许添加模型。<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

二、严格模式与混杂模式的来源

当年Netscape4（网景公司早期的浏览器）和IE4（微软公司早期的浏览器）实现CSS机制时，并没有遵循W3C提出的标准。Netscape4 提供了糟糕的支持，而IE4 虽然接近标准，但依旧未能完全正确的支持标准。尽管IE 5 修复了IE4 许多的问题，但是依然延续CSS实现中的其它故障（主要是盒模型问题）。

为了保障自己的网站在各个浏览器上显示正确，网页开发者们不得不依据各个浏览器自身的规范来使用css，因此大部分网站的css实现并不符合W3C规范的标准。

然而随着标准一致性越来越重要，浏览器开发商不得不面临一个艰难的抉择：逐渐遵循W3C的标准是前进的方向。但是改变现有的css，完全去遵循标准，会使许多旧网站或多或少受到破坏，如果浏览器突然以正确的方式解析现存的css，陈旧的网站的显示必然会受到影响。所以，所有的浏览器都需要提供两种模式：混杂模式服务于旧式规则，而严格模式服务于标准规则。

1、Doctype作用？严格模式与混杂模式如何区分，如何触发这两种模式？它们有何意义？

(1) 声明位于文档中的最前面，处于 标签之前。告诉浏览器的解析器，用什么文档类型规范来解析这个文档。

(2) 严格模式的排版和 JS 运作模式是以该浏览器支持的最高标准运行。

(3) 在混杂模式中，页面以宽松的向后兼容的方式显示。模拟老式浏览器的行为以防止站点无法工作。

(4) DOCTYPE不存在或格式不正确会导致文档以混杂模式呈现。

(5)加入XMI声明可触发，解析方式更改为IE5.5 拥有IE5.5的bug

2.行内元素有哪些？块级元素有哪些？CSS的盒模型？

块级元素: div p h1 h2 h3 h4 form ul

行内元素: a b br i span input select

Css盒模型:内容, border ,margin, padding

3.xhtml和html有什么区别？

区别是：

XHTML 元素必须被正确地嵌套。

XHTML 元素必须被关闭。

标签名必须用小写字母。

XHTML 文档必须拥有根元素。

4.对WEB标准以及W3C的理解与认识？

标签闭合、标签小写、不乱嵌套、提高搜索引擎、使用外链css和js脚本、结构行为表现的分离、文件下载与页面速度更快、内容能被更多的用户所访问、内容能被更广泛的设备所访问、更少的代码和组件，容易维护、改版方便，不需要变动页面内容、提供打印版本而不需要复制内容、提高网站易用性。

5.前端页面有哪三层构成，分别是什么？作用是什么？

Html结构层，css表示层，js行为层

作用是：实现分离。使用html去创建文档的结构，使用css去设置文档的呈现效果，使用js脚本去实现文档的行为。

6.CSS引入的方式有哪些？link和@import的区别是？

内联, 内嵌, 外链, 导入.

区别：同时加载

前者无兼容性，后者CSS2.1以下浏览器不支持.

Link 支持使用javascript改变样式，后者不可.

7.CSS选择符有哪些？哪些属性可以继承？优先级算法如何计算？内联和important哪个优先级高？

标签选择符 类选择符 id选择符

继承不如指定 Id>class>标签选择

后者优先级高

8.你做的页面在哪些浏览器测试过？这些浏览器的内核分别是什么？

ie(ie内核) 火狐 (Gecko) 谷歌 (webkit) opear(Presto)

9.css的基本语句构成是？

选择器{属性1:值1;属性2:值2;.....}

10.写出几种IE6 BUG的解决方法

a.双边距BUG float引起的 使用display

f.Min-height 最小高度！ Important 解决'

h.为什么没有办法定义1px左右的宽度容器（IE6默认的行高造成的，使用over:hidden,zoom:0.08 line-height:1px）

11.浏览器标准模式和怪异模式之间的区别是什么？

盒子模型 渲染模式的不同

12.你如何对网站的文件和资源进行优化？期待的解决方案包括：

文件合并

文件最小化/文件压缩

缓存的使用

使用雪碧图

13.标签上title与alt属性的区别是什么？

Alt 当图片不显示是 用文字代表。Title 为该属性提供信息

15.解释css sprites，如何使用。

Css 精灵 把一堆小的图片整合到一张大的图片上，减轻服务器对图片的请求数量。

16.清除浮动的几种方式，各自的优缺点？

a.使用空标签清除浮动 clear:both（理论上能清除任何标签，增加无意义的标签）

b.使用overflow:auto（空标签元素清除浮动而不得不增加无意义代码的弊端，使用zoom:1用于兼容IE）

c.是用afert伪元素清除浮动(用于非IE浏览器)

17.什么是语义化的HTML，语义化的作用是什么？

语义化的HTML就是写出的HTML代码，符合内容的结构化（内容语义化），选择合适的标签（代码语义化），能够便于开发者阅读和写出更规范性代码的同时让浏览器的爬虫和机器很好地解析。

作用：

- a.有利于SEO，有利于搜索引擎爬虫更好的理解我们的网页，从而获取更多的有效信息，提升网页的权重。
- b.在没有CSS的时候能够清晰的看出网页的结构，增强可读性。
- c.便于团队开发和维护，语义化的HTML可以让开发者更容易的看明白，从而提高团队的效率和协调能力。
- d.支持多终端设备的浏览器渲染。

19.如何优化代码？

- a.代码重用
- b.避免全局变量（封闭空间，模块化）
- c.拆分函数避免函数过于臃肿
- d.注释
- e.代码压缩
- f.减少http请求
- g.图片css sprite（雪碧图）

20.什么是媒体查询，如何使用？

媒体查询可以让我们根据设备显示器的特性（如视口宽度、屏幕比例、设备方向：横向或纵向）为其设定CSS样式，媒体查询由媒体类型和一个或多个检测媒体特性的条件表达式组成。媒体查询中可用于检测的媒体特性有 width、height 和 color（等）。使用媒体查询，可以在不改变页面内容的情况下，为特定的一些输出设备定制显示效果。

一，首先需要在html文档中添加以下代码，用来兼容移动设备的显示效果

```
<meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=1,user-scalable=no"/>
```

注意：width=device-width:宽度等于当前设备的宽度

initial-scale=1：初始的缩放比例（默认为1）

maximum-scale=1：允许用户缩放到的最大比例（默认为1）

user-scalable=no：用户不能手动缩放

二，在CSS文件中写CSS响应式媒体查询

例：

```
@media screen and (max-width:720px) and (min-width:320px){  
  
    body{  
  
        background-color:red;  
  
    }  
}
```

```
@media (max-width:320px){

    body{

        background-color:blue;

    }

}
```

该段媒体查询的意思是：当设备屏幕宽度在320px——720px之间时，媒体查询中body的背景色（background-color:red;）会重叠之前的body背景色，当设备屏幕宽度在320px以下时，媒体查询中body的背景色（background-color:blue;）会重叠之前的body背景色

怎样才能做到屏幕适配？

需要用的是：

```
<meta name="viewport" content="width=device-width">
```

meta viewport 中有6个通用属性：

- 1) width 设置layout viewport的宽度 正整数或字符串 'width-device'
- 2) initial-scale 设置页面的初始缩放值，数字或小数
- 3) minimum-scale 允许用户的最小缩放值 数字或小数
- 4) maximum-scale 允许用户的最大缩放值 数字或小数
- 5) height 设置layout viewport 的高度，这个属性很少用到
- 6) user-scalable 是否允许用户进行缩放 'no'或'yes' 还有2个需要特别注意的属性

21.请说出三种减低页面加载时间的方法

- a、压缩css、js文件
- b、合并js、css文件，减少http请求
- c、外部js、css文件放在最底下
- d、减少dom操作，尽可能用变量替代不必要的dom操作

22.iframe有那些缺点？

- a.会产生很多页面，不容易管理。
- b.iframe框架结构有时会让人感到迷惑，如果框架个数多的话，可能会出现上下、左右滚动条，会分散访问者的注意力，用户体验度差。
- c.代码复杂，无法被一些搜索引擎索引到，这一点很关键，现在的搜索引擎爬虫还不能很好的处理iframe中的内容，所以使用iframe会不利于搜索引擎优化。
- d.很多的移动设备（PDA 手机）无法完全显示框架，设备兼容性差
- e.iframe框架页面会增加服务器的http请求，对于大型网站是不可取的。

23.http和https有什么区别？

在URL前加https://前缀表明是用SSL加密的。 你的电脑与服务器之间收发的信息传输将更加安全。 Web服务器启用SSL需要获得一个服务器证书并将该证书与要使用SSL的服务器绑定。

http和https使用的是完全不同的连接方式,用的端口也不一样,前者是80,后者是443。

http的连接很简单,是无状态的,...

HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议,要比http协议安全

24.浏览器缓存有几种，有什么区别？

网站后台缓存 是网站程序产生的一些临时文件，如PHP程序最喜欢。这个文件一般是在网站程序目录。

服务器缓存是服务器为了提升速度而产生的一些临时文件，如gzip的文件，一般是防在特定的目录。

浏览器缓存是用户本地浏览器缓存的网页文件，比如我们第一次打开一个网站速度会慢一些，第二次打开就很快，就是缓存在了本地。

25.有做过页面性能优化么？如何做的。

- 把样式表置于顶部
- 把脚本置于页面底部
- 避免使用 CSS 表达式 (Expression)
- 使用外部 JavaScript 和 CSS
- 削减 JavaScript 和 CSS
- 用 <link> 代替 @import
- 避免使用滤镜
- 剔除重复脚本
- 减少DOM访问
- 开发智能事件处理程序
- Cookie:
- 减小Cookie体积
- 对于页面内容使用无cookie域名
- 图片:
- 优化图像
- 优化CSS Sprite
- 不要在HTML中缩放图像
- favicon.ico要小而且可缓存

26.XML和JSON的区别？

(1).数据体积方面。

JSON相对于XML来讲，数据的体积小，传递的速度更快些。

(2).数据交互方面。

JSON与JavaScript的交互更加方便，更容易解析处理，更好的数据交互。

(3).数据描述方面。

JSON对数据的描述性比XML较差。

(4).传输速度方面。

JSON的速度要远远快于XML。

27.CSS3有哪些新特性？

CSS3实现圆角 (border-radius) ， 阴影 (box-shadow) ， 对文字加特效 (text-shadow、) ， 线性渐变 (gradient) ， 旋转 (transform) ， transform:rotate(9deg) scale(0.85,0.90) translate(0px,-30px) skew(-9deg,0deg);//旋转,缩放,定位,倾斜，增加了更多的CSS选择器 多背景 rgba，在CSS3中唯一引入的伪元素是::selection.，媒体查询，多栏布局，border-image

28.html5有哪些新特性、移除了那些元素？如何处理HTML5新标签的浏览器兼容问题？如何区分HTML 和 HTML5？

b.语义化更好的内容标签 (header,nav,footer,aside,article,section)

c.音频、视频API(audio,video)

d.画布(Canvas) API

f.本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失；

g. sessionStorage 的数据在浏览器关闭后自动删除

h.表单控件，、date、time、email、url、search

对可用性产生负面影响的元素：frame，frameset，noframes；

如何区分：DOCTYPE声明\新增的结构元素\功能元素

Javascript

1.javascript的typeof返回哪些数据类型

Object number function boolean underfind

2.例举3种强制类型转换和2种类型转换？

强制 (parseInt,parseFloat,number)

3.split() join() 的区别

前者是切割成数组的形式，后者是将数组转换成字符串

4.数组方法pop() push() unshift() shift()

Push()尾部添加 pop()尾部删除

Unshift()头部添加 shift()头部删除

5.call和apply的区别？

相同点:两个方法产生的作用是完全一样的

不同点:方法传递的参数不同

Object.call(this,obj1,obj2,obj3)调用一个对象的一个方法，以另一个对象替换当前对象

Object.apply(this,arguments)应用某一对象的一个方法，用另一个对象替换当前对象。

6.IE和DOM事件监听的区别

- a.执行顺序不一样、
- b.参数不一样
- c.事件加不加on
- d.this指向问题，IE中指向window

7.IE和标准下有哪些兼容性的写法

```
Var ev = ev || window.event
document.documentElement.clientWidth || document.body.clientWidth
Var target = ev.srcElement||ev.target
```

8.添加 删除 替换 插入到某个接点的方法

```
obj.appendChild()  
obj.insertBefore()  
obj.replaceChild()  
obj.removeChild()
```

11.””和“=”的不同

前者是判断值是否相等,后者是判断值及类型是否完全相等。

12.编写一个数组去重的方法

```
function oSort(arr)  
{  
    var result ={};  
    var newArr=[];  
    for(var i=0;i<arr.length;i++)  
    {  
        if(!result[arr])  
        {  
            newArr.push(arr)  
            result[arr]=1  
        }  
    }  
    return newArr  
}
```

12.数组排序

```
var arr [ 1, 2, 3, 4, 5];  
arr.sort(function (a,b) {  
  
    // return a-b; //正序  
    // return b-a; //降序  
  
    return a-b;  
})
```

12.对象转成数组

```

var data={a:1,b:2,c:3};
function toArr(obj){
    var arr1=[];
    for(var i in obj){
        arr1.push(obj[i]);
    }
    return arr1;
    // arr1.fitter(function(x){console.log(arr1(x))});
}
console.log(toArr(data));

```

13.事件绑定和普通事件有什么区别？

事件绑定就是针对dom元素的事件，绑定在dom元素上

普通事件即为非针对dom元素的事件；

例如：

普通事件

```

var btn = document.getElementById("hello");
btn.onclick = function(){
    alert(1);
};
btn.onclick = function(){
    alert(2);
};
//这个事件只会弹出
2;

```

事件绑定

```

var btn = document.getElementById("hello");
btn.addEventListener("click",function(){
    alert(1);
},false);
btn.addEventListener("click",function(){
    alert(2);
},false);
//这个事件首先会弹出1，
然后在弹出2;

```

13.es6新特性

箭头操作符；对class的支持（constructor构造函数）；不定参数...x；let和const关键字；for of遍历；模块的支持import；promise异步函数的处理模式(pending等待中；resolve返回成功，reject返回失败)；

14.b继承a的方法?

b.prototype=new a;

15.document load 和document ready的区别?

Document.onload 是在结构和样式加载完才执行js

Document.ready原生种没有这个方法，jquery中有 \$.ready(function)

15.null和undefind的区别

null是表示一个空的对象，转为数值为0，undefind表示一个空的原始值，转为数值为NAN

undefind指本该有一个值，但却并无定义，null表示没有对象，不应该有值

16.JavaScript 的事件流模型都有什么?

事件流描述的是从页面中接收事件的顺序。DOM 结构是树形结构，当页面中的某个元素触发了某个事件，事件会从最顶层的 window 对象开始，向下传播到目标元素，途径的祖先节点都会触发对应的事件，如果当前节点的该事件绑定了事件处理函数的话，则会执行该函数当事件达到目标元素并执行绑定函数（如果有绑定的话）后，事件又会向上传播到 window 元素，途径的祖先节点都会触发对应的事件（如果绑定事件处理函数的话）

17.BOM 对象有哪些，列举 window 对象?

window 对象，是 JS 的最顶层对象，其他的 BOM 对象都是 window 对象的属性。

location 对象，浏览器当前URL信息。

navigator 对象，浏览器本身信息。

screen 对象，客户端屏幕信息。

history 对象，浏览器访问历史信息。

18.请简述 AJAX 及基本步骤?

AJAX即“Asynchronous Javascript And XML”（异步 JavaScript 和 XML），是指一种创建交互式网页应用的网页开发技术。通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

AJAX 基本步骤：

a.初始化ajax对象

b.连接地址，准备数据

c.发送请求

- d.接收数据（正在接收，尚未完成）
- E.接收数据完成

19.HTTP 状态消息 200 302 304 403 404 500 分别表示什么？

200：请求已成功，请求所希望的响应头或数据体将随此响应返回。

302：请求的资源临时从不同的 URI 响应请求。由于这样的重定向是临时的，客户端应当继续向原有地址发送以后的请求。只有在 Cache-Control 或 Expires 中进行了指定的情况下，这个响应才是可缓存的。

304：如果客户端发送了一个带条件的 GET 请求且该请求已被允许，而文档的内容（自上次访问以来或者根据请求的条件）并没有改变，则服务器应当返回这个状态码。304 响应禁止包含消息体，因此始终以消息头后的第一个空行结尾。

403：服务器已经理解请求，但是拒绝执行它。

404：请求失败，请求所希望得到的资源未被在服务器上发现。

500：服务器遇到了一个未曾预料的状态，导致了它无法完成对请求的处理。一般来说，这个问题都会在服务器端的源代码出现错误时出现。

20.GET和POST的区别，何时使用POST？

GET和POST的区别：

GET：一般用于查询数据，使用URL传递参数，由于浏览器对地址栏长度有限制，所以对使用get方式所发送信息的数量有限制，同时浏览器会记录（历史记录，缓存）中会保留请求地址的信息，包括地址后面的数据。get 只能发送普通格式（URL 编码格式）的数据。

POST：一般用于向服务器发送数据，对所发送的数据的大小理论上是没有限制，浏览器会缓存记录地址，但是不会记录 post 提交的数据。post 可以发送纯文本、URL编码格式、二进制格式的字符串，形式多样。

一般在这种情况下会使用post：

- a.以提交为目的的请求（类似语义化，get 表示请求，post 表示提交）。
- b.发送私密类数据（用户名、密码）（因为浏览器缓存记录特性）。
- c.向服务器发送大量数据（数据大小限制区别）。
- d.上传文件图片时（数据类型区别）。

21.AJAX有哪些局限性？

b.安全问题 AJAX 暴露了与服务器交互的细节。

c.对搜索引擎的支持比较弱。不会执行你的 JS 脚本，只会操作你的网页源代码。

22.new 操作符的工作原理？

当使用 new 操作符调用构造函数，函数实际会经历如下步骤：

- 创建一个新对象。
- 把函数中上下文（作用域）对象this指向该对象。
- 执行代码，通过this给新对象添加属性或方法。
- 返回对象。

23.null 和 undefined 的区别？

- null： null表示空值，转为数值时为0。
- undefined： undefined表示“缺少值”，就是此处应该有一个值，但是还没有定义。
- 变量被声明了，但没有赋值时，就等于undefined。
- 对象没有赋值的属性，该属性的值为undefined。
- 函数没有返回值时，默认返回undefined。

24.JavaScript 原型，原型链？有什么特点？

JavaScript原型：每创建一个函数，函数上都有一个属性为 prototype，它的值是一个对象。这个对象的作用在于当使用函数创建实例的时候，那么这些实例都会共享原型上的属性和方法。

原型链：在 JavaScript 中，每个对象都有一个指向它的原型（prototype）对象的内部链接（proto）。这个原型对象又有自己的原型，直到某个对象的原型为 null 为止（也就是不再有原型指向）。这种一级一级的链结构就称为原型链（prototype chain）。当查找一个对象的属性时，JavaScript 会向上遍历原型链，直到找到给定名称的属性为止；到查找到达原型链的顶部（Object.prototype），仍然没有找到指定的属性，就会返回 undefined。

25.cookie 和 session 有什么区别？

Cookie 存在客户端，session 存在服务端，浏览器通过cookie 向服务端传数据，cookie 有 大小和 数量的限制。

26.json 和 jsonp 是什么，有什么区别。

Json 是 轻量级的数据交换格式，jsonp 是跨域的一个实现方法，通过动态创建script 的方式，请求远程的一个脚本，执行本地的callback，把数据用参数的方式传进执行的函数里。

27.谈谈你对webpack的看法(了解即可)

WebPack 是一个模块打包工具，你可以使用WebPack管理你的模块依赖，并编译输出模块们所需的静态文件。它能够很好地管理、打包Web开发中所用到的HTML、Javascript、CSS以及各种静态文件（图片、字体等），让开发过程更加高效。对于不同类型的资源，webpack有对应的模块加载器。webpack模块打包器会分析模块间的依赖关系，最后 生成了优化且合并后的静态资源。

29.什么是版本控制

版本控制是一种记录一个或若干文件内容变化，以便将来查阅修改以及更新。

28.说说你对作用域链的理解

作用域链的作用是保证执行环境里有权访问的变量和函数是有序的，作用域链的变量只能向上访问，变量访问到window对象即被终止，作用域链向下访问变量是不被允许的。

29.Javascript垃圾回收方法

标记清除 (mark and sweep)

这是JavaScript最常见的垃圾回收方式，当变量进入执行环境的时候，比如函数中声明一个变量，垃圾回收器将其标记为“进入环境”，当变量离开环境的时候（函数执行结束）将其标记为“离开环境”。

垃圾回收器会在运行的时候给存储在内存中的所有变量加上标记，然后去掉环境中的变量以及被环境中变量所引用的变量（闭包），在这些完成之后仍存在标记的就是要删除的变量了

引用计数(reference counting)

在低版本IE中经常会出现内存泄露，很多时候就是因为其采用引用计数方式进行垃圾回收。引用计数的策略是跟踪记录每个值被使用的次数，当声明了一个变量并将一个引用类型赋值给该变量的时候这个值的引用次数就加1，如果该变量的值变成了另外一个，则这个值得引用次数减1，当这个值的引用次数变为0的时候，说明没有变量在使用，这个值没法被访问了，因此可以将其占用的空间回收，这样垃圾回收器会在运行的时候清理掉引用次数为0的值占用的空间。

在IE中虽然JavaScript对象通过标记清除的方式进行垃圾回收，但BOM与DOM对象却是通过引用计数回收垃圾的，也就是说只要涉及BOM及DOM就会出现循环引用问题。

30.说说你对闭包的理解

使用闭包主要是为了设计私有的方法和变量。闭包的优点是可以避免全局变量的污染，缺点是闭包会常驻内存，会增大内存使用量，使用不当很容易造成内存泄露。在js中，函数即闭包，只有函数才会产生作用域的概念

闭包有三个特性：

- a.函数嵌套函数
- b.函数内部可以引用外部的参数和变量
- c.参数和变量不会被垃圾回收机制回收

31.javascript继承的4种方法

- 1, 原型链继承
- 2, 借用构造函数继承
- 3, 组合继承(原型+借用构造)
- 4, 原型式继承

32.创建ajax的过程(原生js中的ajax, 用的少可以了解下)

- (1)创建 XMLHttpRequest 对象,也就是创建一个异步调用对象.
 - (2)创建一个新的 HTTP 请求,并指定该 HTTP 请求的方法、URL 及验证信息.
 - (3)设置响应 HTTP 请求状态变化的函数.
 - (4)发送 HTTP 请求.
 - (5)获取异步调用返回的数据.
 - (6)使用JavaScript和DOM实现局部刷新.
- 代码如下:

```
var xmlhttp = new XMLHttpRequest();

xmlhttp.open('GET','demo.php','true');

xmlhttp.send()

xmlhttp.onreadystatechange = function(){

    if(xmlhttp.readyState === 4 & xmlhttp.status === 200){

        }

    }

}
```

33.请解释一下 JavaScript 的同源策略。(了解即可)

概念:同源策略是客户端脚本(尤其是Javascript)的重要的安全度量标准。它最早出自Netscape Navigator2.0,其目的是防止某个文档或脚本从多个不同源装载。

这里的同源策略指的是:协议,域名,端口相同,同源策略是一种安全协议。指一段脚本只能读取来自同一起来源的窗口和文档的属性。

34.ajax的缺点和在IE下的问题？

ajax的缺点

- 1、ajax不支持浏览器back按钮。
- 2、安全问题 AJAX暴露了与服务器交互的细节。
- 3、对搜索引擎的支持比较弱。
- 4、破坏了程序的异常机制。
- 5、不容易调试。

IE缓存问题

在IE浏览器下，如果请求的方法是GET，并且请求的URL不变，那么这个请求的结果就会被缓存。解决这个问题的办法可以通过实时改变请求的URL，只要URL改变，就不会被缓存，可以通过在URL末尾添加上随机的时间戳参数('t' = + new Date().getTime())

35.谈谈你对重构的理解

网站重构：在不改变外部行为的前提下，简化结构、添加可读性，而在网站前端保持一致的行为。也就是是在不改变UI的情况下，对网站进行优化，在扩展的同时保持一致的UI。

对于传统的网站来说重构通常是：

表格(table)布局改为DIV+CSS

使网站前端兼容于现代浏览器(针对于不合规的CSS、如对IE6有效的)

对于移动平台的优化

针对于SEO进行优化

深层次的网站重构应该考虑的方面

减少代码间的耦合

让代码保持弹性

严格按规范编写代码

设计可扩展的API

代替旧有的框架、语言(如VB)

增强用户体验

通常来说对于速度的优化也包含在重构中

压缩JS、CSS、image等前端资源(通常是由服务器来解决)

程序的性能优化(如数据读写)

采用CDN来加速资源加载

对于JS DOM的优化

HTTP服务器的文件缓存

36 常用那几种浏览器测试？有哪些内核(Layout Engine)?

(Q1)浏览器：IE, Chrome, FireFox, Safari, Opera。

(Q2)内核：Trident, Gecko, Presto, Webkit。

说下行内元素和块级元素的区别？行内块元素的兼容性使用？（IE8 以下）

(Q1)行内元素：会在水平方向排列，不能包含块级元素，设置width无效，height无效(可以设置line-height)，margin上下无效，padding上下无效。

块级元素：各占据一行，垂直方向排列。从新行开始结束接着一个断行。

(Q2)兼容性：display:inline-block;*display:inline;*zoom:1;

37.清除浮动有哪些方式？

(Q1)

- (1) 父级div定义height。
- (2) 结尾处加空div标签clear:both。
- (3) 父级div定义伪类:after和zoom。
- (4) 父级div定义overflow:hidden。
- (5) 父级div定义overflow:auto。
- (6) 父级div也浮动，需要定义宽度。
- (7) 父级div定义display:table。
- (8) 结尾处加br标签clear:both。

38.box-sizing常用的属性有哪些？分别有什么作用？

(Q1)box-sizing: content-box|border-box|inherit;

(Q2)content-box:宽度和高度分别应用到元素的内容框。在宽度和高度之外绘制元素的内边距和边框(元素默认效果)。

border-box:元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制。通过从已设定的宽度和高度分别减去边框和内边距才能得到内容的宽度和高度。

39.HTML5 为什么只需要写？

HTML5不基于 SGML，因此不需要对DTD进行引用，但是需要doctype来规范浏览器的行为（让浏览器按照它们应该的方式来运行）。

而HTML4.01基于SGML,所以需要对DTD进行引用，才能告知浏览器文档所使用的文档类型。

40.介绍一下你对浏览器内核的理解？

主要分成两部分：渲染引擎(layout engineer或Rendering Engine)和JS引擎。

渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核。

JS引擎则：解析和执行javascript来实现网页的动态效果。

最开始渲染引擎和JS引擎并没有区分的很明确，后来JS引擎越来越独立，内核就倾向于只指渲染引擎。

哪些常见操作会造成内存泄漏？

内存泄漏指任何对象在您不再拥有或需要它之后仍然存在。

垃圾回收器定期扫描对象，并计算引用了每个对象的其他对象的数量。如果一个对象的引用数量为0（没有其他对象引用过该对象），或对该对象的惟一引用是循环的，那么该对象的内存即可回收。

setTimeout 的第一个参数使用字符串而非函数的话，会引发内存泄漏。

闭包、控制台日志、循环（在两个对象彼此引用且彼此保留时，就会产生一个循环）

41.对WEB标准以及W3C的理解与认识

标签闭合、标签小写、不乱嵌套、提高搜索机器人搜索几率、使用外链css和js脚本、结构行为表现的分离、文件下载与页面速度更快、内容能被更多的用户所访问、内容能被更广泛的设备所访问、更少的代码和组件，容易维护、改版方便，不需要变动页面内容、提供打印版本而不需要复制内容、提高网站易用性；

42.xhtml和html有什么区别

HTML是一种基本的WEB网页设计语言，XHTML是一个基于XML的置标语言

最主要的不同：

XHTML 元素必须被正确地嵌套。

XHTML 元素必须被关闭。

标签名必须用小写字母。

XHTML 文档必须拥有根元素。

43.行内元素有哪些？块级元素有哪些？CSS的盒模型？

块级元素: div p h1 h2 h3 h4 form ul
行内元素: a b br i span input select
Css盒模型:内容, border ,margin, padding

44.前端页面有哪三层构成, 分别是什么?作用是什么?

结构层 Html 表示层 CSS 行为层 js

闭包有三个特性:

- 1.函数嵌套函数
- 2.函数内部可以引用外部的参数和变量
- 3.参数和变量不会被垃圾回收机制回收

45.css3有哪些新特性?

CSS3实现圆角 (border-radius) , 阴影 (box-shadow) ,
对文字加特效 (text-shadow、) , 线性渐变 (gradient) , 旋转 (transform)
transform:rotate(9deg) scale(0.85,0.90) translate(0px,-30px) skew(-9deg,0deg);//旋转,缩放,定位,
倾斜
增加了更多的CSS选择器 多背景 rgba
在CSS3中唯一引入的伪元素是::selection.
弹性盒子, 动画

46.CSS隐藏元素的几种方法 (至少说出三种)

Opacity:元素本身依然占据它自己的位置并对网页的布局起作用。它也将响应用户交互;
Visibility:与 opacity 唯一不同的是它不会响应任何用户交互。此外, 元素在读屏软件中也会被隐藏;
Display:display 设为 none 任何对该元素直接打用户交互操作都不可能生效。此外, 读屏软件也不会读到元素的内容。这种方式产生的效果就像元素完全不存在;
Position:不会影响布局, 能让元素保持可以操作;

47.介绍一下CSS的盒子模型?

有两种, IE 盒子模型、标准 W3C 盒子模型; IE的content部分包含了 border 和 padding;
盒模型: 内容(content)、填充(padding)、边界(margin)、 边框(border).

48.CSS 选择符有哪些? 哪些属性可以继承? 优先级算法如何计算? CSS3新增伪类有那些?

```

id选择器 ( # myid)
类选择器 (.myclassname)
标签选择器 (div, h1, p)
相邻选择器 (h1 + p)
子选择器 (ul > li)
后代选择器 (li a)
通配符选择器 ( * )
属性选择器 (a[rel = "external"])
伪类选择器 (a: hover, li: nth - child)
可继承的样式: font-size font-family color, UL LI DL DD DT;
不可继承的样式: border padding margin width height ;
优先级就近原则, 同权重情况下样式定义最近者为准;
优先级为:
JavaScript
!important > id > class > tag
important 比 内联优先级高

!important > id > class > tag
important 比 内联优先级高

```

49.简述一下src与href的区别

href 是指向网络资源所在位置, 建立和当前元素 (锚点) 或当前文档 (链接) 之间的链接, 用于超链接。

src是指向外部资源的位置, 指向的内容将会嵌入到文档中当前标签所在位置; 在请求src资源时会将其指向的资源下载并应用到文档内, 例如js脚本, img图片和frame等元素。当浏览器解析到该元素时, 会暂停其他资源的下载和处理, 直到将该资源加载、编译、执行完毕, 图片和框架等元素也如此, 类似于将所指向资源嵌入当前标签内。这也是为什么将js脚本放在底部而不是头部。

50.什么是CSS Hack?

一般来说是针对不同的浏览器写不同的CSS,就是 CSS Hack。

IE浏览器Hack一般又分为三种, 条件Hack、属性级Hack、选择符Hack (详细参考CSS文档: css 文档) 。例如:

```

// 1、条件Hack
<!--[if IE]>
  <style>
    .test{color:red;}
  </style>
<![endif]-->
// 2、属性Hack
.test{
  color:#090\9; /* For IE8+ */
  *color:#f00; /* For IE7 and earlier */
}

```



```
_color:#ff0; /* For IE6 and earlier */
}
// 3、选择符Hack
* html .test{color:#090;} /* For IE6 and earlier */
* + html .test{color:#ff0;} /* For IE7 */
```

51.简述同步和异步的区别

同步是阻塞模式，异步是非阻塞模式。

同步就是指一个进程在执行某个请求的时候，若该请求需要一段时间才能返回信息，那么这个进程将会一直等待下去，直到收到返回信息才继续执行下去；

异步是指进程不需要一直等下去，而是继续执行下面的操作，不管其他进程的状态。当有消息返回时系统会通知进程进行处理，这样可以提高执行的效率。

52.px和em的区别

px和em都是长度单位，区别是，px的值是固定的，指定是多少就是多少，计算比较容易。em得值不是固定的，并且em会继承父级元素的字体大小。

浏览器的默认字体高都是16px。所以未经调整的浏览器都符合：1em=16px。那么12px=0.75em，10px=0.625em

53.一次完整的HTTP事务是怎样的一个过程？

基本流程：

域名解析

建立TCP连接后发起http请求

服务器端响应http请求，浏览器得到html代码

浏览器解析html代码，并请求html代码中的资源

浏览器对页面进行渲染呈现给用户

54.写一个function，清除字符串前后的空格。（兼容所有浏览器）

```
function trim(str) {
    if (str && typeof str === "string") {
        return str.replace(/(^\\s*)|(\\s*)$/g, ""); //去除前后空白符
    }
}
```

55.对前端工程师这个职位你是怎么样理解的？

- a. 前端是最贴近用户的程序员，前端的能力就是能让产品从 90分进化到 100 分，甚至更好
- b. 参与项目，快速高质量完成实现效果图，精确到1px；
- c. 与团队成员，UI设计，产品经理的沟通；
- d. 做好的页面结构，页面重构和用户体验；
- e. 处理hack，兼容、写出优美的代码格式；
- f. 针对服务器的优化、拥抱最新前端技术。

56.前端模块化

在JavaScript发展初期就是为了实现简单的页面交互逻辑，寥寥数语即可；如今CPU、浏览器性能得到了极大的提升，很多页面逻辑迁移到了客户端（表单验证等），随着web2.0时代的到来，Ajax技术得到广泛应用，jQuery等前端库层出不穷，前端代码日益膨胀

这时候JavaScript作为嵌入式的脚本语言的定位动摇了，JavaScript却没有为组织代码提供任何明显帮助，甚至没有类的概念，更不用说模块（module）了，JavaScript极其简单的代码组织规范不足以驾驭如此庞大规模的代码

57.模块

既然JavaScript不能handle如此大规模的代码，我们可以借鉴一下其它语言是怎么处理大规模程序设计的，在Java中有一个重要带概念——package，逻辑上相关的代码组织到同一个包内，包内是一个相对独立的王国，不用担心命名冲突什么的，那么外部如果使用呢？直接import对应的package即可

```
import java.util.ArrayList;
```

遗憾的是JavaScript在设计时定位原因，没有提供类似的功能，开发者需要模拟出类似的功能，来隔离、组织复杂的JavaScript代码，我们称为模块化。

一个模块就是实现特定功能的文件，有了模块，我们就可以更方便地使用别人的代码，想要什么功能，就加载什么模块。模块开发需要遵循一定的规范，各行其是就都乱套了

规范形成的过程是痛苦的，前端的先驱在刀耕火种、茹毛饮血的阶段开始，发展到现在初具规模，简单了解一下这段不凡的历程

函数封装

我们在讲函数的时候提到，函数一个功能就是实现特定逻辑的一组语句打包，而且JavaScript的作用域就是基于函数的，所以把函数作为模块化的第一步是很自然的事情，在一个文件里面编写几个相关函数就是最开始的模块了

```
function fn1(){  
    statement  
}  
  
function fn2(){  
    statement  
}
```

这样在需要的以后夹在函数所在文件，调用函数就可以了

这种做法的缺点很明显：污染了全局变量，无法保证不与其他模块发生变量名冲突，而且模块成员之间没什么关系。

58.AMD

AMD 即Asynchronous Module Definition，中文名是异步模块定义的意思。它是一个在浏览器端模块化开发的规范

由于不是JavaScript原生支持，使用AMD规范进行页面开发需要用到对应的库函数，也就是大名鼎鼎RequireJS，实际上AMD 是 RequireJS 在推广过程中对模块定义的规范化的产出

requireJS主要解决两个问题

多个js文件可能有依赖关系，被依赖的文件需要早于依赖它的文件加载到浏览器

js加载的时候浏览器会停止页面渲染，加载文件越多，页面失去响应时间越长

59.CMD

CMD 即Common Module Definition通用模块定义，CMD规范是国内发展出来的，就像AMD有个requireJS，CMD有个浏览器的实现SeaJS，SeaJS要解决的问题和requireJS一样，只不过在模块定义方式和模块加载（可以说运行、解析）时机上有所不同

60.AMD与CMD区别

最明显的区别就是在模块定义时对依赖的处理不同

AMD推崇依赖前置，在定义模块的时候就要声明其依赖的模块
CMD推崇就近依赖，只有在用到某个模块的时候再去require

这种区别各有优劣，只是语法上的差距，而且requireJS和SeaJS都支持对方的写法

AMD和CMD最大的区别是对依赖模块的执行时机处理不同，注意不是加载的时机或者方式不同

61.谈谈对前端模块化的理解

前端模块化就是把复杂的文件分成一个个独立的模块，比如js文件，分成独立的模块之后有利于代码的重用和维护，但是这样又会引来模块与模块之间的依赖问题，所以就有了CommonJS、AMD、CMD规范

62.前端优化（提高网页的加载速度）

1、使用css sprites，可以有效的减少http请求数 2、使用缓存 3、压缩js，css文件，减小文件体积 4、使用cdn，减小服务器负担 5、懒加载图片 6、预加载css，js文件 7、避免dom结构的深层次嵌套 8、给DOM元素添加样式时，把样式放到类中，直接给元素添加类，减少重构，回流

63.封装cookie的添加，删除，查询方法

cookie是存储在浏览器端的，可以用于存储sessionID，也可以用于自动登陆，记住密码等，但是在浏览器端并没有官方的操作cookie的方法，下面我们来封装一下：

```
CookieUtil= {
  addCookie:function(key,value,options){
    var str=key+"="+escape(value);
    if(options.expires){
      var curr=new Date(); //options.expires的单位是小时
      curr.setTime(curr.getTime()+options.expires*3600*1000);
      options.expires=curr.toGMTString();
    }
    for(var k in options){ //有可能指定了cookie的path, cookie的domain
      str+=";"+k+"="+options[k];
    }
    document.cookie=str;
  },
  queryCookie:function(key){
    var cookies=document.cookie;
    //获得浏览器端存储的cookie,格式是key=value;key=value;key=value
    cookies+=";";
    var start=cookies.indexOf(key);
    if(start<=-1){ return null; } //说明不存在该cookie
    var end=cookies.indexOf(";",start);
    var value=cookies.slice(start+key.length+1,end);
    return unescape(value);
  },
  deleteCookie:function(key){
    var value=CookieUtil.queryCookie(key);
    if(value===null){return false;}
    CookieUtil.addCookie(key,value,{expires:0}); //把过期时间设置为0, 浏览器
    会马上自动帮我们删除cookie
  }
}
```

}

64.如何实现浏览器内多个标签页之间的通信?

调用localStorage、cookies等本地存储方式

65.CSS居中（包括水平居中和垂直居中）

内联元素居中方案

水平居中设置:

1.行内元素

设置 text-align:center;

2.Flex布局

66.请你谈谈Cookie的弊端

cookie虽然在持久保存客户端数据提供了方便，分担了服务器存储的负担，但还是有很多局限性的。第一：每个特定的域名下最多生成20个cookie

1.IE6或更低版本最多20个cookie

2.IE7和之后的版本最后可以有50个cookie。

3.Firefox最多50个cookie

4.chrome和Safari没有做硬性限制

IE和Opera 会清理近期最少使用的cookie，Firefox会随机清理cookie。

cookie的最大大约为4096字节，为了兼容性，一般不能超过4095字节。

IE 提供了一种存储可以持久化用户数据，叫做userData，从IE5.0就开始支持。每个数据最多128K，每个域名下最多1M。这个持久化数据放在缓存中，如果缓存没有清理，那么会一直存在。

67.CSS中 link 和@import 的区别是?

A: (1) link属于HTML标签，而@import是CSS提供的;(2) 页面被加载的时，link会同时被加载，而@import引用的CSS会等到页面被加载完再加载;(3) import只在IE5以上才能识别，而link是HTML标签，无兼容问题;(4) link方式的样式的权重 高于@import的权重。

68.为什么要初始化CSS样式。

因为浏览器的兼容问题，不同浏览器对有些标签的默认值是不同的，如果没对CSS初始化往往会出现浏览器之间的页面显示差异。

当然，初始化样式会对SEO有一定的影响，但鱼和熊掌不可兼得，但力求影响最小的情况下初始化。

最简单的初始化方法就是：{padding: 0; margin: 0;}（不建议）

淘宝的样式初始化：

```
body, h1, h2, h3, h4, h5, h6, hr, p, blockquote, dl, dt, dd, ul, ol, li,
pre, form, fieldset, legend, button, input, textarea, th, td { margin:0;
padding:0; }
body, button, input, select, textarea { font:12px/1.5tahoma, arial, \5b8b
\4f53; }
h1, h2, h3, h4, h5, h6{ font-size:100%; }
address, cite, dfn, em, var { font-style:normal; }
code, kbd, pre, samp { font-family:couriernew, courier, monospace; }
small{ font-size:12px; }
ul, ol { list-style:none; }
a { text-decoration:none; }
a:hover { text-decoration:underline; }
sup { vertical-align:text-top; }
sub{ vertical-align:text-bottom; }
legend { color:#000; }
fieldset, img { border:0; }
button, input, select, textarea { font-size:100%; }
table { border-collapse:collapse; border-spacing:0; }
```

69.浮动元素引起的问题和解决办法？

浮动元素引起的问题：

- (1) 父元素的高度无法被撑开，影响与父元素同级的元素
- (2) 与浮动元素同级的非浮动元素会跟随其后
- (3) 若非第一个元素浮动，则该元素之前的元素也需要浮动，否则会影响页面显示的结构

解决方法：使用CSS中的clear:both;属性来清除元素的浮动可解决2、3问题，对于问题1，添加如下样式，给父元素添加clearfix样式：

```
.clearfix:after{content: ".";display: block;height: 0;clear: both;visibil
ity: hidden;}
.clearfix{display: inline-block;} /* for IE/Mac */
```

70.清除浮动的几种方法：

1, 额外标签法,

(缺点：不过这个办法会增加额外的标签使HTML结构看起来不够简洁。)

2, 使用after伪类

```
#parent:after{
  content:".";
  height:0;
  visibility:hidden;
  display:block;
  clear:both;
}
```

3,浮动外部元素

4,设置 `overflow` 为 `hidden` 或者 `auto`

71.请说出三种减少页面加载时间的方法。

1.优化图片

2.图像格式的选择（GIF：提供的颜色较少，可用在一些对颜色要求不高的地方）

3.优化CSS（压缩合并css，如margin-top,margin-left...）

4.网址后加斜杠（如www.campr.com/目录，会判断这个“目录是什么文件类型，或者是目录。”）

5.标明高度和宽度（如果浏览器没有找到这两个参数，它需要一边下载图片一边计算大小，如果图片很多，浏览器需要不断地调整页面。这不但影响速度，也影响浏览体验。

当浏览器知道了高度和宽度参数后，即使图片暂时无法显示，页面上也会腾出图片的空位，然后继续加载后面的内容。从而加载时间快了，浏览体验也更好了。）

6.减少http请求（合并文件，合并图片）。

72.new操作符具体干了什么呢？

1、创建一个空对象，并且 this 变量引用该对象，同时还继承了该函数的原型。

2、属性和方法被加入到 this 引用的对象中。

3、新创建的对象由 this 所引用，并且最后隐式的返回 this 。

```
var obj  = {};  
obj.__proto__ = Base.prototype;  
Base.call(obj);
```

73.哪些操作会造成内存泄漏？

内存泄漏指任何对象在您不再拥有或需要它之后仍然存在。

垃圾回收器定期扫描对象，并计算引用了每个对象的其他对象的数量。如果一个对象的引用数量为0（没有其他对象引用过该对象），或对该对象的惟一引用是循环的，那么该对象的内存即可回收。

setTimeout 的第一个参数使用字符串而非函数的话，会引发内存泄漏。

闭包、控制台日志、循环（在两个对象彼此引用且彼此保留时，就会产生一个循环）

网站重构的理解？

网站重构：在不改变外部行为的前提下，简化结构、添加可读性，而在网站前端保持一致的行为。也就是是在不改变UI的情况下，对网站进行优化，在扩展的同时保持一致的UI。

74.其他问题：

说一些你遇到过的解决兼容的问题

怎么跨越，**get**请求，**post**请求

原来公司工作流程是怎样的，如何与其他人协作的？如何跨部门合作的？

你遇到过比较难的技术问题是？你是如何解决的？

常使用的库有哪些？常用的前端开发工具？开发过什么应用或组件？

你有用过哪些前端性能优化的方法？

http状态码有那些？分别代表是什么意思？

一个页面从输入 **URL** 到页面加载显示完成，这个过程中都发生了什么？

除了前端以外还了解什么其它技术么？你最厉害的技能是什么？

对前端界面工程师这个职位是怎么样理解的？它的前景会怎么样？

你移动端前端开发的理解？（和 **Web** 前端开发的主要区别是什么？）

你对加班的看法？

说说最近最流行的一些东西吧？常去哪些网站？

简单描述一下你做过的项目研发流程？

你认为怎样才是前端工程师？

最近在学什么东西？

你的优点是什么？缺点是什么？

最近在学什么？能谈谈你未来**3**，**5**年给自己的规划吗？

75.Vue.js是什么？

Vue.js不是一个框架，因为它只聚焦视图层，是一个构建数据驱动的Web界面的库。

Vue.js通过简单的API（应用程序编程接口）提供高效的数据绑定和灵活的组件系统。

76.Vue.js它有哪些特性呢？

1.轻量级的框架

2.双向数据绑定

3.指令

4.插件化

Vue.js与其他框架的区别？

1.与AngularJS的区别

相同点：

1.都支持指令：内置指令和自定义指令。

2.都支持过滤器：内置过滤器和自定义过滤器。

3.都支持双向数据绑定。

4.都不支持低端浏览器。

不同点：

1. AngularJS的学习成本高，而Vue.js本身提供的API都比较简单、直观。
2. 在性能上，AngularJS依赖对数据检查，所以Watcher越多越慢。
3. Vue.js使用基于依赖追踪的观察并且使用异步队列更新。所有的数据都是独立触发的。

2.与React的区别

相同点：

1. React采用特殊的JSX语法，Vue.js在组件开发中也推崇编写.vue特殊文件格式，对文件内容都有一些约定，两者都需要编译后使用。

补充：SX就是Javascript和XML结合的一种格式。React发明了JSX，利用HTML语法来创建虚拟DOM。当遇到<，JSX就当HTML解析，遇到{就当JavaScript解析。

78.为什么使用vue？用在哪里？

vue两大特点：响应式编程、组件化

vue的优势：轻量级框架、简单易学、双向数据绑定、组件化、视图、数据和结构的分离、虚拟DOM、运行速度快

vue一般用来开发单页web项目

79.AngularJS与Angular的区别

Angular2.0之前的版本（1.x）叫做AngularJS

1.x的使用是引入AngularJS的js文件到网页。

2.0之后，就是完全不同了。

Angular2.x与Angular1.x 的区别类似 Java 和 JavaScript

80. React的特点

声明式编码

组件化编码

支持客户端与服务器渲染

利用jsx创建虚拟dom